



Linux Ubuntu 26.04 Manual Pages on command 'libtracecmd.3'

\$ man libtracecmd.3

LIBTRACECMD(3) libtraceefs Manual LIBTRACECMD(3)

NAME

libtracecmd - trace-cmd library APIs

SYNOPSIS

```
#include <trace-cmd.h>
```

Open and close trace file:

```
struct tracecmd_input *tracecmd_open(const char *file, int flags);  
struct tracecmd_input *tracecmd_open_fd(int fd, int flags);  
struct tracecmd_input *tracecmd_open_head(const char *file, int flags);  
void tracecmd_close(struct tracecmd_input *handle);  
void tracecmd_set_private(struct tracecmd_input *handle, void *data);  
void *tracecmd_get_private(struct tracecmd_input *handle);
```

Read tracing records from a trace file:

```
int tracecmd_init_data(struct tracecmd_input *handle);  
struct tep_record *tracecmd_read_cpu_first(struct tracecmd_input *handle, int cpu);  
struct tep_record *tracecmd_read_data(struct tracecmd_input *handle, int cpu);  
struct tep_record *tracecmd_read_at(struct tracecmd_input *handle, unsigned long long offset, int *cpu);  
void tracecmd_free_record(struct tep_record *record);  
struct tep_handle *tracecmd_get_tep(struct tracecmd_input *handle);
```

Iterating over events in a trace file:

```
int tracecmd_iterate_events(struct tracecmd_input *handle,
                           cpu_set_t *cpus, int cpu_size,
                           int (*callback)(struct tracecmd_input *,
                                           struct tep_record *,
                                           int, void *),
                           void *callback_data);

int tracecmd_iterate_events_multi(struct tracecmd_input **handles,
                                  int nr_handles,
                                  int (*callback)(struct tracecmd_input *,
                                                  struct tep_record *,
                                                  int, void *),
                                  void *callback_data);

int tracecmd_iterate_events_reverse(struct tracecmd_input *handle,
                                    cpu_set_t *cpus, int cpu_size,
                                    int (*callback)(struct tracecmd_input *,
                                                    struct tep_record *,
                                                    int, void *),
                                    void *callback_data, bool cont);

int tracecmd_follow_event(struct tracecmd_input *handle,
                          const char *system, const char *event_name,
                          int (*callback)(struct tracecmd_input *,
                                          struct tep_event *,
                                          struct tep_record *,
                                          int, void *),
                          void *callback_data);

int tracecmd_follow_missed_events(struct tracecmd_input *handle,
                                  int (*callback)(struct tracecmd_input *,
                                                  struct tep_event *,
                                                  struct tep_record *,
                                                  int, void *),
                                  void *callback_data);
```

```

struct tracecmd_filter *tracecmd_filter_add(struct tracecmd_input handle,
                                           const char *filter_str, bool neg);

int *tracecmd_iterate_reset(struct tracecmd_input *handle);

```

Read tracing instances from a trace file:

```

int tracecmd_buffer_instances(struct tracecmd_input *handle);

const char *tracecmd_buffer_instance_name(struct tracecmd_input *handle, int indx);

struct tracecmd_input *tracecmd_buffer_instance_handle(struct tracecmd_input *handle, int indx);

```

Handle time stamps from a trace file:

```

unsigned long long tracecmd_get_first_ts(struct tracecmd_input *handle);

void tracecmd_add_ts_offset(struct tracecmd_input *handle, long long offset);

int tracecmd_get_tsc2nsec(struct tracecmd_input *handle, int *mult, int pass[]shift, unsigned long long *offset);

```

Get tracing peer information from a trace file:

```

unsigned long long *tracecmd_get_traceid(struct tracecmd_input *handle);

int tracecmd_get_guest_cpumap(struct tracecmd_input *handle, unsigned long long trace_id, const char **name,
int *vcpu_count, const int **cpu_pid);

```

Mapping host and guest trace files:

```

int tracecmd_map_vcpus(struct tracecmd_input **handles, int nr_handles);

struct tracecmd_cpu_map *tracecmd_get_cpu_map(struct tracecmd_input *handle, int cpu);

struct tracecmd_cpu_map *tracecmd_map_find_by_host_pid(struct tracecmd_input *handle,
                                                       int host_pid);

int tracecmd_map_get_host_pid(struct tracecmd_cpu_map *map);

struct tracecmd_input *tracecmd_map_get_guest(struct tracecmd_cpu_map *map);

void tracecmd_map_set_private(struct tracecmd_cpu_map *map, void *priv);

void *tracecmd_map_get_private(struct tracecmd_cpu_map *map);

```

Control library logs:

```

int tracecmd_set_loglevel(enum tep_loglevel level);

```

The libtracecmd(3) library provides APIs to read, parse and write trace-cmd.dat(5) files, recorded with trace-cmd(1) application and containing tracing information from ftrace, the official Linux kernel tracer.

FILES

trace-cmd.h

Header file to include in order to have access to the library APIs.

-ltracecmd

Linker switch to add when building a program that uses the library.

SEE ALSO

libtraceevent(3) libtracefs(3) trace-cmd(1) trace-cmd.dat(5)

AUTHOR

Steven Rostedt <rostedt@goodmis.org[1]>

Tzvetomir Stoyanov <tz.stoyanov@gmail.com[2]>

REPORTING BUGS

Report bugs to <linux-trace-devel@vger.kernel.org[3]>

LICENSE

libtracecmd is Free Software licensed under the GNU LGPL 2.1

RESOURCES

<https://git.kernel.org/pub/scm/utils/trace-cmd/trace-cmd.git/>

COPYING

Copyright (C) 2020 VMware, Inc. Free use of this software is granted under the terms of the GNU Public License (GPL).

NOTES

1. rostedt@goodmis.org

<mailto:rostedt@goodmis.org>

2. tz.stoyanov@gmail.com

<mailto:tz.stoyanov@gmail.com>

3. linux-trace-devel@vger.kernel.org

<mailto:linux-trace-devel@vger.kernel.org>

libtracefs

01/22/2026

LIBTRACECMD(3)